Time Series Analysis of Cross-Listed Stocks

Sanketh Kokkodu Balakrishna and Lopamudra Pal
Department of Computer Science
University of Massachusetts Amherst, MA, 01003
{skokkodubala,lpal}@cs.umass.edu

1 Introduction

Stock Market prediction is a well-known problem in the Financial and Statistical research areas. This time series forecasting problem is of tremendous interest to researchers because of the profit potential it presents. The Efficient-Market Hypothesis (EMH) suggests that stock prices reflect all currently available information and any price change that is not based on newly revealed information is inherently unpredictable. The various factors which influence the stock market are political events, concurrent economic conditions, trader's expectations, changing interest rates, varying foreign exchange rates and many more.
However, researchers in technical analysis believe that stock prices can be predicted from historical analysis because of the assumption that stock prices move in trends and that the information which affect prices enters the market over a finite period, not instantaneously. A lot of interesting work has been carried out in applying Machine Learning algorithms for analyzing price patterns and predicting stock prices and index changes.
In this paper, we present applications of various Machine Learning(ML) Regression algorithms along with their performance for different stocks. Unlike previous work done on predicting stock prices in a single exchange, the focus of this project is to predict the price of a stock listed on the National Stock Exchange(NSE) of India based on historical data collected from NSE and the New York Stock Exchange(NYSE). We hypothesize here that the prices in emerging stock markets like India follow the prices of the developed economies due to the huge impact of businesses in the US. The reason for this hypothesis is the observation of the movement of cross-listed stocks on both the markets.

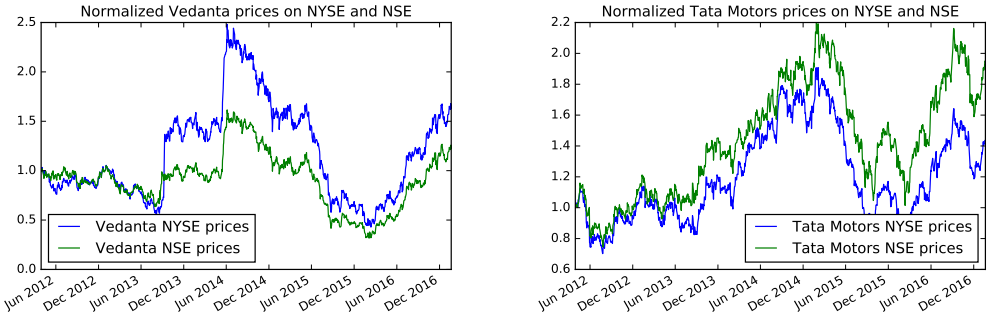


Figure 1: Prices of 'Vedanta' and 'Tata Motors' on NYSE and NSE

Figure.1 shows the normalized prices of the cross-listed stocks 'Vedanta' and 'Tata Motors' on NSE and NYSE. We observe similar patterns and trends on both the markets. Hence, we picked only cross-listed stocks which have a large market share in the US and India for this analysis. For this problem, we collected data from Yahoo Finance for cross listed stocks in different industries like Banking (ICICI),Automotive (TATA MOTORS) and Energy (VEDANTA). Apart from the inherent features (Date, Open, High, Low, Volume, Adjusted Close, Close) in the datasets, we have added

additional features which are described in the Data Sets section. We observed differences in the predicted results, which are described in detail in Section 5.

2 Related Work

Although a lot of work has been done in using Machine Learning to predict stock market prices based on historical data, there is very little literature about the problem we're trying to solve. Some related work done on analyzing stock markets is briefly described below.

2.1 Machine Learning in Prediction

The application of different Machine Learning techniques to predict the stock market is briefly described in [1]. The paper discusses the application of Support Vector Machines, Linear Regression, Prediction using Decision Stumps, Expert Weighting, and Online Learning on two stocks – Google and Yahoo (data collected from Yahoo Finance). It also describes useful parameters and indicators to recognize patterns in stock prices for successful prediction. The author has added an indicator function "Exponential Moving Average(EMA)" to the dataset and observed the behavior of the stocks. From the experiments, the author concludes that using Support Vector Machine combined with Boosting gives satisfactory results.
A detailed study of how textual analysis from newspapers can be used to predict the stock prices with a twenty minutes lagging time is shown in [7]. Using a support vector machine (SVM) derivative specially tailored for discrete numeric prediction and models containing different stock-specific variables, they show that the model containing both article terms and stock price at the time of article release had the best performance in closeness to the actual future stock price (MSE 0.04261), the same direction of price movement as the future price (57.1% directional accuracy) and the highest return using a simulated trading engine (2.06% return).
[4] does a survey and evaluation of the recent developments in stock market prediction models and the effects of global events on stock prices. Comparing various prediction models, they found that Neural Networks(NNs) offer the ability to predict market directions more accurately than other existing techniques. The ability of NNs to learn nonlinear relationships from the training input/output pairs enables them to model non-linear dynamic systems such as stock markets more precisely. Using a good web-mining technique to identify qualitative factors can help improve the prediction accuracy using Neural Networks.

2.2 Technical Indicators in stock market prediction

[9] and [10] present a variety of technical indicators for stock market prediction and propose a new modular genetic programming for finding attractive and statistically sound technical patterns for stock trading. The paper mentions that if technical patterns are complex, they generally match a few cases which degrades the generality if they are highly profitable. Hence it proposes a genetic algorithm to determine the most attractive technical indicators for a dataset which focuses on three criteria: high profitability, simplicity, and frequency within the dataset. [10] gives a brief description of the different classes of technical indicators studied in stock market prediction problem. Technical indicators are classified in two wide groups: oscillators and trend followers. The oscillators are designed to vary at the same time the price varies. They represent a form of the price over a fixed past period, which is the interval used to calculate the indicator. Trend followers work better when markets are immersed in a clear trend. They are not efficient in moments of side market, without a defined trend. This paper centers on the optimization of two indicators, an oscillator- Relative Strength Index (RSI), and a trend follower- Moving Average of Convergence-Divergence (MACD).

2.3 Analysis of Emerging stock markets

Work done on using Artificial Neural Networks (ANN) to predict Istanbul Stock Exchange (ISE) market index value is presented in [6]. The features considered to build the model are previous day's index value, previous day's TL/USD exchange rate, previous day's overnight interest rate and 5 dummy variables each representing the working days of the week. Research showed that fewer the number of hidden layers in the ANN model, the more accurate the predicted are. The ANN performance was compared to the moving averages over a specified past period (5 and 10 days in this

study and it was found that the former outperformed the latter in moving averages).
[2] analyses the results of applying hybrid ML models based on Genetic Algorithms(GA) and Support Vector Machines(SVM) for stock market prediction. The research work done in [2] is on Indian Stocks.
[5] and [8] gives us an overview of planning and visualizing the data and results for our experiments. [5] shows visualizations of how sensitive the Indian Stock Market is on external factors like Mumbai Train bombings, economic factors like the GDP growth rate, Capital inflow in sectors (eg. Construction, Health care, IT). [8] describes a very innovative approach of designing a multisensory human perceptual tool for the real-world task domain of stock market trading. We hypothesize here that the Indian market(NSE) is heavily dependent on some of the developed markets such as the New York Stock Exchange(NYSE).

2.4 Study of co-related stocks

The main idea of the research in [2] is to compute the correlation between stocks and determine the most correlated stocks. Different technical indicators are calculated for these correlated stocks. Using the proposed Genetic Algorithm, feature selection is carried out to find the most important features, which in turn are then given as inputs to Support Vector Machine and the predicted results are analyzed. The original input features are scaled into the range of [1,1] to independently normalize each feature component to the specified range. It ensures the larger value input attributes do not overwhelm smaller value inputs, and thus helps to reduce prediction errors. Neuro-genetic stock prediction system based on financial correlation between companies is explored in [3].

3 Data Sets

3.1 Data Description

Stocks considered in this paper are restricted to firms listed on both the NYSE and NSE only. The data is collected from Yahoo finance (**https://finance.yahoo.com/**) and each dataset is pre-processed individually. Pre-processing includes removing rows corresponding to days when the stock did not trade, merging features from NSE and NYSE listings, creating new features and appending the gold standard values. We consider three stocks of different sectors- ICICI Bank from the Banking sector, Tata Motors from the Automotive sector and Vedanta Limited from the Energy sector. The date range considered for the training data is March 23 2012 to January 23 2017. For testing data, the dates considered are from January 26 2017 to April 26 2017. The total number of training examples after pre-processing reduces to 1210 for the 5 year period and the testing data to 57 data cases for the three month period. We believe that this is sufficient test data to accurately measure prediction accuracy, as the deviation in the data is large enough.

3.2 Feature Description

From the data downloaded directly, each listed stock has has these six features:

1. Open price: The price at which the stock opened on a particular day. This may be different than the close price of the previous day due to after hours trading and difference between supply and demand.
2. High price: The highest price the stock reached for a particular day. This is typically higher than the opening and the closing price.
3. Low price: The lowest price the stock reached for a particular day. This is typically lower than the opening and the closing price.
4. Close price: This is the last traded price for the day. The price may change later to account for post-trading activities.
‘ **5. Volume**: The number of shares traded on a particular day. Indicates the activity and also may indicate the rise or fall of the price.
6. Adjusted Close: The stock's closing price that has been amended to include any distributions and corporate actions that occurred at any time prior to next day's open.
These features are considered for both NSE and NYSE to yield 12 features. Two additional features are defined to help model price movement:

7. Rolling Mean: Rolling Mean or Simple Moving Average(SMA) is the arithmetic moving average calculated over k previous days. For the current day, SMA is given as

$$\frac{P_N + P_{N-1} + P_{N-2} + P_{N-3} + ... + P_{N-k}}{k}$$

where $P_N ... P_{N-k}$ are the prices for the previous k days. We have considered k = 5 in our problem.
8. Daily Return: The amount by which a stock has moved from the previous day. It is given by $\frac{P_N}{P_{N-1}} - 1$, where P_N is the current day price and P_{N-1} is the previous day closing price.
These additional features are added to both the stock listings to give a total of 16 features. The y_{pred} value is taken as the following day's closing price of the stock.

4 Methodology

In this section, the complete pipeline which we have built is described. The pipeline is divided into sections which are detailed as separate subsections.

4.1 Tools Used

We used a variety of tools and software to build different parts. The complete pre-processing was done in the 'pandas' library, which offered large number of methods and flexibility in handling the data. 'Python' was used for the complete pipeline and sci-kit learn provided regression methods and hyper parameter and feature optimization.

4.2 Pre-processing

The data obtained from Yahoo Finance had a lot of missing values and 'nan' values when the stock did not trade. We dropped those rows which had incomplete information. The features for the stocks on the NSE and NYSE were combined to get the data set for one cross-listed stock. On some days, the stock traded on one exchange but did not on the other exchange. Those rows are dropped for getting consistent results. The Y_{pred} values i.e the closing price of the following day is also appended to each feature vector to create a proper format for sk-learn fit and predict methods. Two additional features are added as described in the Data Sets section. Since only 16 features are used which are core to the stock's price movement, no feature engineering or selection is required in the pre-processing stage.

4.3 Machine Learning models

Some of the models which we tried are detailed below. The results of these models are described in the Results section.
• **Linear Regression** : Linear Regression is a parametric regression method that assumes the relationship between y and x is a linear function with parameters $w = [w_1, ...w_D]^T$ and b.The regression function is given as $f_{Lin}(X) = \sum_{d=1}^{D} w_d x_d + b$. This is our baseline method.
• **Decision Tree Regressor** : Decision Tree Regressor is a parametric regressor that outputs data cases using a conjunction of rules organized into a binary tree structure. Each node in the tree consists of a rule of the form $(xd < t)$ or $(xd = t)t$. The simplicity of this model is the primary reason we picked this model.

• **Ridge Regression:** In regression methods such as least squares linear regression, the parameters are susceptible to very high variance.To control variance, we might need to regularize the coefficients. Ridge Regression is a form of regularized least squares when the weights are penalized using the l_2 norm ${||w||_2}^2 = w^T w = \sum_{d=1}^{D} {w_d}^2$.
The regularization of weight parameters during learning is achieved by setting w^* as

$$w^* = argmin_w \frac{1}{N} \sum_{i=1}^{N} (y_i - x_i w)^2 + \lambda {||w||_2}^2$$

$$= argmin_w \frac{1}{N} \sum_{i=1}^{N} (y_i - x_i w)^2 ... st {||w||_2}^2 <= c$$

The optimized regularized weights which is the ridge regression estimator is $w^* = (X^T X + \lambda I)^{-1} X^T Y$. The regularization of Linear Regression may help Ridge to model the non-linearity of the stock prices. Hence, we chose this model.

• **K Neighbors Regressor**: The KNN regressor is a non-parametric regressor that stores the training data, and outputs each new instance according to a majority vote over its set of K nearest neighbors in the training set. The regression function is given as $f_{KNN}(x) = \frac{1}{K} \sum_{i \in N_k(x)} y_i$. K[the number of neighbors considered] is the major hyper-parameter to be set in this classifier. We considered this model because the training time of this model is very less.

• **Random Forest Regressor** : Random forests regressors are ensemble learning methods for regression that operate by constructing a multitude of decision trees at training time and outputting the mean prediction of the individual trees. Random decision forests correct for decision trees' habit of over-fitting to their training set. This ensemble method may give better results than decision trees.

• **Elastic Nets** : The elastic net method overcomes the limitations of the Ridge and Lasso method which uses a penalty function based on $\beta_1 = \sum_{j=1}^{p} |\beta_j| . \beta_1 = \sum_{j=1}^{p} \beta_j ..$
The estimates from the elastic net method are defined by $\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1) . \hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1$. In this, both l1 and l2 norm is regularized. This variation may help Elastic Nets model the prices accurately. Hence, we chose this model.

• **MLPRegressor** : A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. The activation functions are described by $y(v_i) = \tanh(v_i)$ and $y(v_i) = (1 + e^{-v_i})^{-1}$. Almost all the projects which are related report their best results using Neural Networks. The lack of data in our case may not necessarily give us the best results. However, the ability of Neural Networks to learn non-linear relationships made us choose this model.

4.4 Pipeline

The flow chart of our pipeline is shown in Figure.2. The data is collected and pre-processed as described above. The Feature Selection step helps us filter out unnecessary features. We then create a pipeline combining the Regression and the Feature Selection models. This pipeline is then used in a Hyper Parameter Selection method called GridSearchCV, which performs an exhaustive search of the different hyper-parameters and selects the best results by cross-validation. Finally the model is trained on the training data, after which we predict on the testing data. We evaluate the model using the Root Mean Squared Error. Evaluation is also done on how well the predictions fit the gold standard values. The model having the least RMSE and the best fit is identified as the best model. The RMSE is given as

$$RMSE(f, D) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i))^2}$$

5 Experiments and Results

We experimented with all the models described in the methodology section. Some of the models clearly outperformed others.

5.1 Baseline

The baseline model we used is Linear Regression. The deviation of predictions from the actuals and the line plot of the prediction vs actual closing price for our testing data is shown in Fig.3.

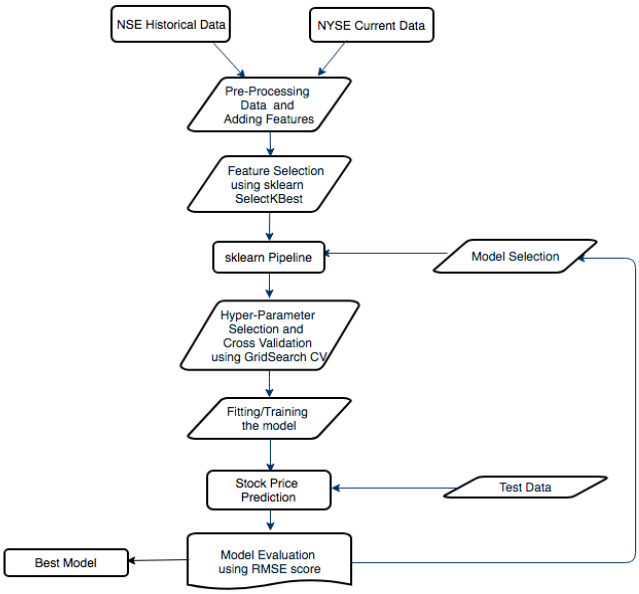


Figure 2: Block Diagram of Pipeline Implementation

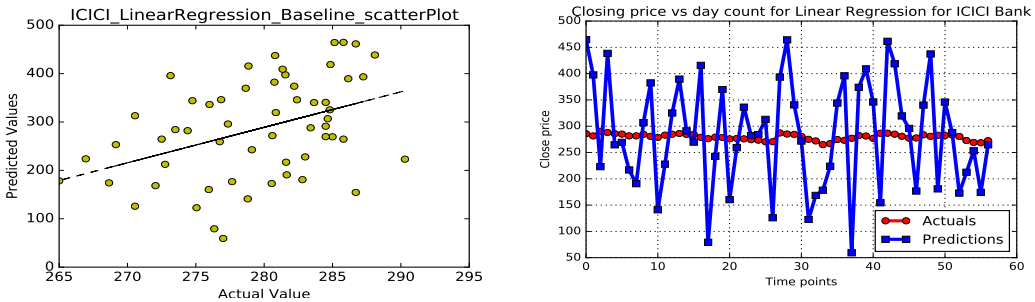


Figure 3: Predicted vs actual price for Linear Regression

The scatterplot of the Y and $\hat{Y}$ values is shown in the first sub-plot. The deviation from the linear line indicates how far the predicted value is from the actual value. The RMSE obtained from the baseline method without any feature selection and hyperparameter optimization is 99.97.
Analysis: The predicted values always have a deviation from the gold standard due to high prediction variance of the model. Linear Regression fits a straight line through the data points, and fails to consider the non-linear nature of the stock prices. Hence, Linear Regression only finds the best fitting straight line, giving rise to such errors.

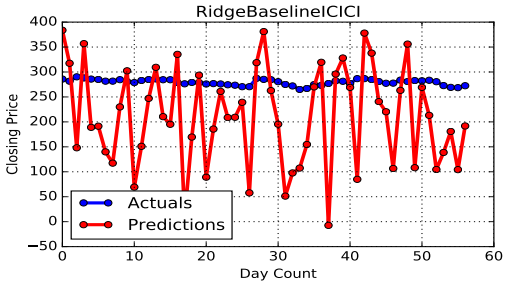
5.2 Model Evaluation

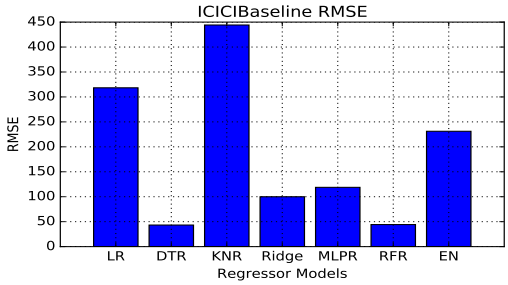
Different models are evaluated based on accuracy and speed on all the three data sets. The results on the ICICI stock data are reported below.

5.2.1 Accuracy

The accuracy of different models on the ICICI stock data are as follows:
Decision Tree(DTR): Decision tree gives an RMSE of 43.287. With feature selection and hyperparameter optimization, the RMSE reduces to 6.97.
Ridge Regression(Ridge): Ridge regressor gives an RMSE of 118.84. With feature selection and hyperparameter optimization, the RMSE reduces to 4.53.
KNN(KNR): KNN gives an RMSE of 318.3. With feature selection and hyperparameter selection,

the RMSE reduces to 4.35.
Random Forest(RFR):Random Forest gives an RMSE of 44.22. With feature selection and hyperparameter selection, the RMSE reduces to 6.12.
Elastic Nets(EN): Elastic Nets gives an RMSE of 231.2. With feature selection and hyperparameter selection, the RMSE reduces to 4.4.
Neural Network(MLPR): The MLP regressor model is also evaluated on the data. The RMSE recorded with and without feature selection and hyperparameter optimization is 127 and 4.9 respectively.
Analysis: Although Elastic Nets gave the least accuracy, the variance of the actual values are modelled better by Ridge Regressor. The reason for Elastic Nets and Ridge giving similar results is due to the fact that Ridge is l1 normalization and Elastic Net combines l1 and l2 penalties. All the models were also tested on the Tata Motors and Vedanta stocks, and similar results were obtained. The line plot of predicted values vs the actual gold standard values for the Ridge Regressor and the RMSE values for all the models is shown in Figure.4.



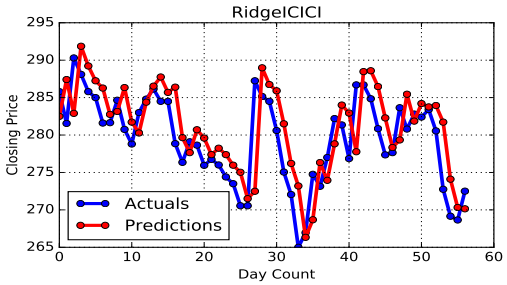


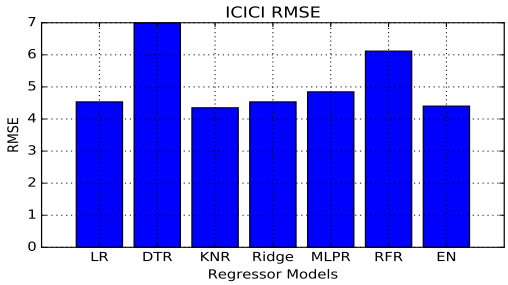
Actuals vs Predictions

Accuracies of different models on ICICI stock data

Figure 4: Model performance before feature selection and optimization

The line plot for Ridge Regressor and the accuracy of all the models with feature selection and hyperparameter optimization on ICICI stock data is summarized in Figure.5.





Actuals vs Predictions

Accuracies of different models on ICICI stock data

Figure 5: Model performance after feature selection and optimization

5.2.2 Value Optimization

Although we see a huge improvement in the predicted values after feature selection and hyperprameter optimization, the output of the regressors seem to be shifted by δ on the majority of the predictions. We propose the following method to account for the shift
Correction: The data is trained on the training data as before. We reduce the testing data by one case and hold out the case as the correction sample x_c. The rest of the testing data is used for testing. x_c is then used to shift the output by δ for all the predictions. The predicted output is now given as

$$\hat{y} = y' - (x_c - y_c)$$

where y_c is the gold standard output for the correction sample. After the correction, the Ridge regressor is found to perform exceedingly well, giving an RMSE of 0.33.

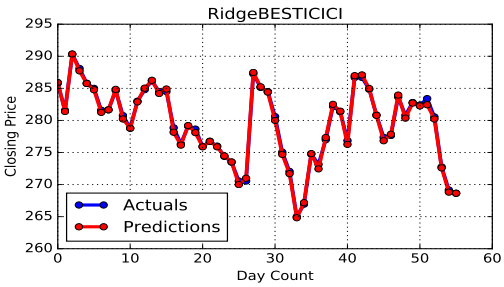


Figure 6: Price movement after Value optimization

Analysis: After the correction is performed, the model tends to perform exceedingly well, and the predicted prices accurately modelled. This method works well only for ICICI data. Different types of corrections may be needed for different stocks, based on the level of shift. Hence, we conclude that one model and one method will not work well for all the stocks, and feature engineering and method selection is needed for individual stocks.

5.2.3 Speed

Speed is one of the factors we need to consider when evaluating stock prediction models in High Frequency Trading(HFT). While we do not deal with HFT in our project, we do need to consider speed if we're applying our method to a large number of listings since we'll have only a 8 hour window to compute predictions before the NSE opens the next day. The training and test computation time for the ICICI stock is summarized in the Table 1. The time for different stocks are comparable to these times.

Model Names	Stock	Speed in seconds
DecisionTreeRegressor	ICICI	0.3531
ElasticNet	ICICI	0.1716
KNeighborsRegressor	ICICI	0.1489
MLPRegressor	ICICI	2.8148
RandomForestRegressor	ICICI	5.5945
Ridge	ICICI	0.0494

Table 1: Table of speed listings for ICICI Stock

It is observed that Ridge regression performs best in terms of speed. The entire training and testing of all the data cases takes 0.04 s to complete, which is sufficient for the scope of the problem.

6 Discussion and Conclusion

We observe that certain trade-offs have to be made in each model in terms of speed or accuracy. Ridge regression gives the overall best accuracy for all the stocks and also gives the best results in terms of speed. Since we don't consider speed to be a big factor in our problem, we conclude that Ridge regression is the best performing model for our problem. Most of the related work treats stock prediction as a classification problem. [1] achieves a classification accuracy of 60% using Support Vector Machines. Most of the papers reported their best results using Neural Network. The reason why neural network failed for our problem is because of the low number of data samples. After the correction step, our model predicted near-perfect prediction values for the ICICI data but failed to generalize for other stocks. We did not address the classification problem of predicting whether the price increases or decreases for a particular day, but it is an interesting problem to pursue further. Also, we would like to factor the impact that news has on a movement. This can be achieved by scrapping data off news websites, predicting sentiment of the news and modelling it in the price movement prediction. We conclude that the stock regression problem is a hard problem to generalize but with sufficient data and with the consideration of only technical influencers, the price of a particular stock can be accurately predicted.

References

[1]Vatsal H. Shah. *Machine Learning Techniques for Stock Prediction. 2009*,

[2]Rohit Choudhry, and Kumkum Garg. *A hybrid machine learning system for stock market forecasting. In: World Academy of Science, Engineering and Technology, p. 39 (2008)*,

[3]Yung-Keun Kwon, Sung-Soon Choi, Byung-Ro Moon. *Stock prediction based on financial correlation, Proceedings of the 2005 conference on Genetic and evolutionary computation, June 25-29, 2005, Washington DC, USA [doi>10.1145/1068009.1068351]*,

[4]Paul D. Yoo, Maria H. Kim, Tony Jan.*Machine learning techniques and use of event information for stock market predictions: A survey and evaluation" in , Sydney, Australia:University of Technology, 2005.*,

[5]Suchismita Naik.*A Narrative Data Visualization Of The Indian Stock Market, Proceedings of the 8th Indian Conference on Human Computer Interaction, December 07 - 09, 2016, Mumbai, India [doi>10.1145/3014362.3014382]*,

[6]Birgul Egeli,Meltem Ozturan,Bertan Badur,*Stock market prediction using artificial neural networks, in: Hawaii International Conference on Business, 2003.*,

[7]Robert P. Schumaker, Hsinchun Chen. *Textual analysis of stock market prediction using breaking financial news: The AZFin text system, ACM Transactions on Information Systems (TOIS), v.27 n.2, p.1-19, February 2009 [doi>10.1145/1462198.1462204]*

[8]K. V. Nesbitt, S. Barrass, *"Finding trading patterns in stock market data", IEEE Comput. Graph. Appl., vol. 24, no. 5, pp. 45-55, Sep. 2004.*

[9]Seung-Kyu Lee , Byung-Ro Moon, *A new modular genetic programming for finding attractive technical patterns in stock markets, Proceedings of the 12th annual conference on Genetic and evolutionary computation, July 07-11, 2010, Portland, Oregon, USA [doi>10.1145/1830483.1830704]*

[10]Diego J. Bodas-Sagi , Pablo Fernández , J. Ignacio Hidalgo , Francisco J. Soltero , José L. Risco-Martín, *Multiobjective optimization of technical market indicators, Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, July 08-12, 2009, Montreal, Québec, Canada [doi>10.1145/1570256.1570266]*